Pelletron Digital Filtering

Local application

Wed, Mar 28, 2007

Some Pelletron signals available for ECool are hard to analyze. A bump may exist, but buried in enough noise that it is hard to see. A new LA called `PELF` is designed to use a digital filtering scheme to tease out the bumps amid the noise. This note describes the algorithm used.

The signals to be treated in this way are the following:

`R:` *names*	*Description*	*Range*
`ACCRV`	Accel tube current	± 200 μA
`DECRV`	Decel tube current	± 200 μA
`COLRV`	Column current	± 200 μA
`GVMVV`	GVM out volts	± 300 kV
`TPSERV`	TPS error volts	± 10 V
`CPO1V`	CPO #1 volts	± 100 kV

Corresponding digital filter output signals are the same but with names ending in `A`, not `V`.

The digital filter used is a Bessel Lowpass type of order 2. The following algorithm is used:

```
Local double Filter(double xarg, double *y)
/*    Perform digital filter algorithm for input channel value */
      {
      double  x;

      x = xarg / GAIN;
      y[0] = y[1];
      y[1] = y[2];
      y[2] = x + c1 * y[0] + c2 * y[1];
      return (y[2]);
      }
```

where the following constants are used for a 15 Hz sample rate:

```
#define GAIN (368.8278871) /* GAIN factor in filter algorithm */
#define c1 (-0.9118429230) /* coefficients used in algorithm */
#define c2 ( 1.9091316313)
```

As implemented in the program, the input values are given in A/D volts, in the range ± 10 V. The algorithm keeps track of the two previous ouput values in computing the current output value.

The resulting output data, in raw units of ±32768 range, is deposited into the reading fields of a series of consecutive analog channels.

Here is the parameter layout for `PELF`:

Prompt		*Size*	*Meaning*
`ENABLE`	`B`	2	Enable Bit# for this LA
`INPUT`	`C`	2	Input signals base Chan#
`NCHANS`		2	#channels
`OUTPUT`	`C`	2	Output signals base Chan#

IRM execution time is approximately `(45 + 30*n)` μs, with `n` the number of channels specified.